Claims

1. An apparatus for generating computer assembly code, comprising:

an abstract routine generator for receiving a data stream comprising a multimedia routine and for outputting a generic abstract representation thereof; and

a translator for said abstract routine generator for receiving said abstract representation and for outputting processor specific code for processing multimedia input data.

2. The apparatus of Claim 1, where in said abstract routine generator builds an abstract routine during runtime.

3. The apparatus of Claim 1, wherein said abstract routine generator builds an abstract routine in the form of a graph.

4. The apparatus of Claim 1 wherein said multimedia data comprise SIMD input data.

5. The apparatus of Claim 1, wherein said multimedia data comprise image input data.

6. The apparatus of Claim 1, wherein said multimedia data comprise audio input data.

7. The apparatus of Claim 3, wherein said graph is input to said translator.

8. The apparatus of Claim 3, wherein the output of said translator is in assembly code.

9. The apparatus of Claim 1, wherein said translator's configuration can be changed by use of a dynamic library link.

10. The apparatus of Claim 1, wherein said processor-specific code performs any of the operations of add, sub, multiply, average, maximum, minimum, compare, and, or, xor, pack, unpack, and merge on said input data.

11. The apparatus of Claim 3, wherein said graph is a function of any of source block, target block, change in the block, color, stride, change in stride, display block, and spatial filtering.

12. A method for generating assembly code, comprising:

providing an abstract routine generator for generating a generic abstract representation of an input stream, and input comprising multimedia a routine; and

providing a translator for receiving said abstract representation from said abstract routine generator and for outputting processor-specific code for processing multimedia input data.

13. The method of Claim 12, wherein said abstract routine generator builds the abstract routine during runtime.

14. The method of Claim 13, wherein said abstract routine is a graph.

15. The method of Claim 12, wherein said multimedia input data comprise SIMD data.

16. The method of Claim 12, said multimedia input data comprise image data.

17. The method of Claim 12, wherein said multimedia input data comprise audio data.

18. The method of claim 14, wherein said graph is input to said translator.

19. The method of claim 12, wherein the output of said translator is assembly code.

20. The method of Claim 12, wherein said processor-specific code performs any of the operations of add, sub, multiply, average, maximum, minimum, compare, and, or, xor, pack, unpack, and merge on said multimedia input data.

21. The method of Claim 14, wherein said graph is a function of any of source block, target block, change in the block, color, stride, change in stride, display block, and spatial filtering.

22. The method of Claim 12, wherein said translator can be changed by use of a dynamic library link.